

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

THIS PAGE BLANK (USPTO)



JFW

01807.101570.

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
YOUENN FABLET)	Examiner: Not Yet Assigned
Application No.: 10/790,847)	Group Art Unit: Not Yet Assigned
Filed: March 3, 2004)	
For: METHOD OF OFFERING A)	
SERVICE PROVIDED BY A)	
SERVER COMPUTER IN A)	
COMMUNICATION NETWORK)	May 27, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

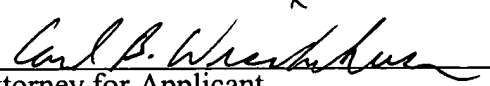
Sir:

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed is a certified copy of the following French application:

0302557 filed March 3, 2003.

Applicant's undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,


Attorney for Applicant

Registration No. 43,279

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200



THIS PAGE BLANK (USPTO)



BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le **01 MARS 2004**

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

A handwritten signature in black ink, appearing to read 'M. Planche', enclosed within a large, loopy oval stroke.

Martine PLANCHE

**INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE**

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

THIS PAGE BLANK (USPTO)



26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

BREVET D'INVENTION CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI

cerfa
N° 11354*03


REQUÊTE EN DÉLIVRANCE page 1/2

BR1

Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 W / 210502

REMISE DES PIÈCES DATE 3 MARS 2003 LIEU 75 INPI PARIS N° D'ENREGISTREMENT 0302557 NATIONAL ATTRIBUÉ PAR L'INPI DATE DE DÉPÔT ATTRIBUÉE 0 3 MARS 2003 PAR L'INPI		1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE SANTARELLI 14, avenue de la Grande Armée 75017 PARIS	
Vos références pour ce dossier (facultatif) BIF023427/HST/IM			
Confirmation d'un dépôt par télécopie <input type="checkbox"/> N° attribué par l'INPI à la télécopie		2 NATURE DE LA DEMANDE Cochez l'une des 4 cases suivantes	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
<i>Demande de brevet initiale</i> <i>ou demande de certificat d'utilité initiale</i>		N° _____ Date _____ N° _____ Date _____	
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>		<input type="checkbox"/> N° _____ Date _____	
3 TITRE DE L'INVENTION (200 caractères ou espaces maximum) Procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication			
4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE		Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ Date _____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»	
5 DEMANDEUR (Cochez l'une des 2 cases) <input checked="" type="checkbox"/> Personne morale <input type="checkbox"/> Personne physique		CANON KABUSHIKI KAISHA Société de droit Japonais _____ 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, JAPON _____ JAPON JAPONAISE N° de télécopie (facultatif) _____	
Nom ou dénomination sociale _____ Prénoms _____ Forme juridique _____ N° SIREN _____ Code APE-NAF _____ Domicile ou siège Rue _____ Code postal et ville _____ Pays _____ Nationalité _____ N° de téléphone (facultatif) _____ Adresse électronique (facultatif) _____		<input type="checkbox"/> S'il y a plus d'un demandeur, cochez la case et utilisez l'imprimé «Suite»	

REMISE DES PIÈCES DATE LIEU 3 MARS 2003 N° D'ENREGISTREMENT 75 INPI PARIS NATIONAL ATTRIBUÉ PAR L'INPI 0302557		Réservé à l'INPI		DB 540 W / 010801	
Vos références pour ce dossier : (facultatif)			BIF023427/HST/IM		
6 MANDATAIRE (s'il y a lieu)					
Nom					
Prénom					
Cabinet ou Société			SANTARELLI		
N° de pouvoir permanent et/ou de lien contractuel					
Adresse	Rue	14 Avenue de la Grande Armée			
	Code postal et ville	75017 PARIS			
	Pays	01 40 55 43 43			
N° de téléphone (facultatif)					
N° de télécopie (facultatif)					
Adresse électronique (facultatif)					
7 INVENTEUR (S)			Les inventeurs sont nécessairement des personnes physiques		
Les demandeurs et les inventeurs sont les mêmes personnes			<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non : Dans ce cas remplir le formulaire de Désignation d'inventeur(s)		
8 RAPPORT DE RECHERCHE			Uniquement pour une demande de brevet (y compris division et transformation)		
Établissement immédiat ou établissement différé			<input checked="" type="checkbox"/> <input type="checkbox"/>		
Paiement échelonné de la redevance (en deux versements)			Uniquement pour les personnes physiques effectuant elles-mêmes leur propre dépôt <input type="checkbox"/> Oui <input type="checkbox"/> Non		
9 RÉDUCTION DU TAUX DES REDEVANCES			Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Obtenue antérieurement à ce dépôt pour cette invention (joindre une copie de la décision d'admission à l'assistance gratuite ou indiquer sa référence) : AG		
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes					
10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (Nom et qualité du signataire)			VISA DE LA PRÉFECTURE OU DE L'INPI		
Hélène STANKOFF N°98.0601 SANTARELLI					

5 La présente invention concerne un procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication.

Elle concerne également un procédé de test d'accès à un service par un ordinateur client et un procédé de validation d'un message reçu par un ordinateur intermédiaire du réseau de communication.

10 Au sein d'un réseau de communication informatique du type Internet, des ordinateurs serveurs offrent de plus en plus souvent des services à d'autres ordinateurs, appelés ordinateurs clients, de ce réseau de communication.

En pratique, l'ordinateur client envoie des données à l'ordinateur serveur qui les traite et retourne un résultat.

15 De tels services sont appelés services Web.

Un service Web est un service identifié par une URI et accessible via XML et un protocole Internet.

Du fait de l'augmentation de ces services disponibles sur un réseau de communication, les protocoles d'échange de données entre des ordinateurs
20 sont fréquemment standardisés.

Ainsi, le protocole SOAP est un protocole permettant d'échanger des informations structurées au-dessus du réseau Internet.

Selon ce protocole SOAP, les informations échangées sont structurées au moyen de balises XML (acronyme du terme anglais "eXtended
25 Markup Language").

La norme SOAP définit la structure générale des messages échangés ainsi que les traitements devant être réalisés par un ordinateur envoyant ou recevant des messages SOAP.

Ainsi, lors de l'exécution d'un service Web, un ou plusieurs
30 messages comprenant des données en format XML sont échangés sur le réseau de communication.

Le protocole SOAP est un protocole extensible, c'est-à-dire que la norme ne définit que le cœur du protocole, des fonctionnalités supplémentaires pouvant être ajoutées à ce protocole.

5 Ces fonctionnalités supplémentaires sont appelées "features". La norme SOAP version 1.2 propose une convention pour décrire ces fonctionnalités supplémentaires, cette convention reposant sur l'utilisation de propriétés associées à chaque fonctionnalité.

On trouvera une description de la norme SOAP 1.2 à l'adresse électronique suivante :

10 <http://www.w3.org/TR/2002/WD-soap12-part1-20020626/>

En particulier, chaque propriété est décrite par son nom et par son type de données.

La description de chaque fonctionnalité comprend, outre les propriétés, un modèle décrivant comment ces propriétés sont utilisées lors de la mise en œuvre de cette fonctionnalité.

Par ailleurs, on connaît un langage de description d'un service informatique WSDL (acronyme du terme anglais "Web Service Description Language") qui est particulièrement bien adapté à décrire un service sur un réseau de communication.

20 Ce langage WSDL est lui-même une application du langage de balisage XML.

On trouvera une description du langage WSDL 1.1 sur le site informatique à l'adresse <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.

25 Un document électronique de description d'un service en langage WSDL comporte généralement deux parties. Une première partie, appelée "partie abstraite" est adaptée à décrire les messages échangés entre ordinateur du réseau de communication lors de la fourniture d'un service.

30 En particulier, cette première partie permet de définir le type de données échangées, le type de message utilisé lors de l'exécution du service, ainsi que les opérations mises en œuvre, définies par les messages qui sont échangés lors de l'exécution du service.

Le document de description d'un service WSDL comporte également une seconde partie adaptée à définir des informations relatives à la transmission des messages sur le réseau de communication.

5 Cette deuxième partie, appelée "partie concrète" précise notamment via un couplage (binding) le protocole de communication qui est effectivement utilisé pour le transfert des messages, ainsi que le format dans lequel les données seront représentées pour leur communication au sein du réseau de communication.

10 Ainsi les services proposés sur Internet échangent principalement des données XML dont le type est connu à l'avance, par exemple au travers d'une description d'un service tel qu'un document WSDL.

Par ailleurs, de nombreux outils permettent de manipuler des données XML, c'est-à-dire de produire des données XML, de les transformer ou encore d'utiliser ces données XML.

15 On connaît ainsi un langage de transformation de documents XML, du type d'un script XSL (acronyme du terme anglais "eXtensible Stylesheet Language") défini par la norme X3C.

20 En pratique, une unité de traitement XSL prend en entrée un document écrit en langage de balisage XML et un document XSL. Ce dernier comporte un ensemble de règles de transformation qui va permettre la génération d'un nouveau document qui peut être soit au format XML ou dans un format texte. La norme XSL est elle-même basée sur la norme XML.

25 De même, on connaît un modèle d'objet de document (en anglais Document Object Model ou DOM) qui consiste en une interface ou API (neutre au point de vue du langage de programmation) qui permet à des programmes/scripts d'accéder à et modifier un document XML. L'interface DOM est notamment supporté par différents langages de script tel que JavaScript. Outre JavaScript, d'autres langages, tels que Perl ou Python permettent la manipulation de documents XML via l'interface DOM.

30 Les différents outils de traitement permettant de mettre en œuvre ces scripts de traitement de données XML sont de plus en plus souvent intégrés

aux terminaux d'un réseau de communication, au niveau d'un client ou d'un serveur.

Or, les capacités de manipulation de données XML sont différentes d'un terminal à un autre.

5 Par ailleurs, les traitements des données XML peuvent être effectués à différents moments, et notamment par un ordinateur client avant qu'il n'envoie les données XML, par un ordinateur serveur avant qu'il n'utilise les données XML, par un ordinateur serveur avant qu'il n'envoie les données XML ou par un ordinateur client avant qu'il n'utilise les données XML.

10 On distingue ainsi des types principaux de traitements, le pré-traitement de données en format XML, qui aura lieu à la réception des données XML, avant leur utilisation, et le post-traitement de données qui prend place après l'utilisation de données XML, avant leur envoi.

Dès lors, il est intéressant sur un tel réseau de communication qu'un
15 ordinateur client connaisse quel type de traitement un ordinateur serveur offre.

Par ailleurs, certains serveurs exigent l'intégration de certains outils de traitement au niveau des ordinateurs client.

Actuellement, il n'est pas possible de décrire sur le réseau de communication le type de traitement (pré-traitement ou post-traitement)
20 proposé par l'un ou l'autre des terminaux du réseau de communication.

On connaît certains réseaux dans lesquels des traitements sont définis et supportés par un ordinateur serveur de manière prédéfinie, sans que d'autres types de traitement puissent être ajoutés de manière extensive.

On connaît également des bases de données spécifiques qui
25 supportent un traitement XSL. Dans ce cas, un ordinateur client peut envoyer une requête pour un document dans lequel un script XSL est inclus.

Le script XSL est utilisé pour transformer les documents identifiés dans la base de données avant leur transfert à l'ordinateur client. Un tel système est cependant limité à une interaction entre un ordinateur client et une
30 base de données spécifique pour laquelle l'ordinateur client sait que l'outil de traitement XSL sera supporté.

La présente invention a ainsi pour but de résoudre les inconvénients précités et de proposer un système dans lequel il est possible de définir sur le réseau de communication les traitements à effectuer qui produiront ou utiliseront des données en format XML d'un message.

5 A cet effet, la présente invention vise un procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication.

Ce procédé comprend une étape d'envoi d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication.

10 La présente invention permet ainsi de définir au travers d'un document de description d'un service du type WSDL directement les traitements qui peuvent être appliqués aux données en format XML échangées lors de la mise en œuvre du service.

Grâce à la définition d'une fonctionnalité, du type d'une fonctionnalité ("feature") telle que définie dans la norme SOAP, il est possible de décrire et d'utiliser cette fonctionnalité dans le document de description d'un service.

15 L'utilisation d'une telle fonctionnalité permet d'ajouter sans limitation différents traitements pouvant être mis en œuvre en différents terminaux du réseau de communication.

Grâce à l'intégration de cette fonctionnalité dans un document de description d'un service du type WSDL, il est possible de définir et d'informer les différents terminaux du réseau de communication de l'existence de ces traitements dans le réseau.

25 Selon une caractéristique préférée de l'invention, cette fonctionnalité définit les traitements adaptés à produire ou utiliser les données en format XML définies dans une première partie abstraite de document de description d'un service. La fonctionnalité de traitement peut ainsi être offerte pour l'ensemble des protocoles disponibles.

30

Il est ainsi possible de spécifier des traitements applicables à des données XML dont le type est connu et défini au travers d'une description comme un document WSDL.

De préférence, cette fonctionnalité décrivant un traitement adapté à manipuler (utiliser ou produire) des données en format XML abstraites, sa description est de préférence associée à ces données abstraites à l'intérieur du document de description d'un service.

Selon une caractéristique avantageuse de l'invention, le pré-traitement ou le post-traitement des données est mis en œuvre via un langage de script.

Selon un autre mode de réalisation de l'invention, qui permet avantageusement l'utilisation de la fonctionnalité pour tout type de protocole, la fonctionnalité est définie comme une donnée en format XML dans une première partie abstraite du document de description d'un service.

Dans ce mode de réalisation, cette donnée en format XML définissant la fonctionnalité est de préférence encodée dans une seconde partie concrète du document de description d'un service.

Ainsi, la fonctionnalité est proposée au niveau abstrait du document WSDL. Dans ce cas, le traitement ajouté se traduit comme une donnée XML (optionnelle ou obligatoire) ajoutée au message. Cette donnée XML est ensuite encodée au niveau de la partie concrète du document WSDL comme tout autre partie du message abstrait.

Ainsi, la fonctionnalité est offerte à tous les protocoles.

Selon une autre caractéristique préférée de l'invention, la description de la fonctionnalité comprend une liste de propriétés supportées par la fonctionnalité, les propriétés définissant au moins respectivement :

- le nœud du réseau de communication adapté à exécuter le traitement ; et
- le type de traitement.

Cette fonctionnalité peut comporter ainsi différentes informations définies sous la forme de propriétés.

Il est notamment possible de définir le type de traitement et l'endroit du réseau de communication adapté à mettre en œuvre ce traitement.

Cette fonctionnalité peut comprendre une propriété adaptée à spécifier si le traitement est effectué à l'envoi ou à la réception du message.

5 De préférence, cette fonctionnalité comprend en outre une propriété adaptée à spécifier le message ou un ensemble de messages sur lequel s'applique le traitement.

De même, cette fonctionnalité comprend de préférence une propriété adaptée à définir les données produites ou utilisées par le traitement, et
10 éventuellement le type de ces données.

Il est ainsi possible grâce au type de ces données définies dans le document de description WSDL de valider avant et après traitement les données générées en déterminant si le type de ces données correspond effectivement au type prédéfini au travers de la description de la fonctionnalité.

15 Selon une autre caractéristique préférée de l'invention, cette fonctionnalité comprend en outre une propriété adaptée à spécifier si le traitement est effectué à l'envoi ou à la réception du message, afin de déterminer s'il s'agit d'un post-traitement ou d'un pré-traitement des données en format XML du message.

20 Selon une caractéristique avantageuse, la description de la fonctionnalité comprend également une propriété adaptée à spécifier si le traitement à effectuer est obligatoire ou optionnel.

En pratique, pour au moins une propriété supportée par la fonctionnalité, la description de cette fonctionnalité comprend une liste de
25 valeurs attribuables à cette propriété.

On peut ainsi utiliser le mécanisme défini par la norme SOAP pour décrire les différentes propriétés de cette fonctionnalité.

Selon un autre aspect de l'invention, elle concerne un procédé de test d'accès à un service par un ordinateur client d'un réseau de
30 communication, à partir d'un document de description d'un service.

Ce procédé de test d'accès comprend les étapes suivantes :

- extraction d'une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication ;

- 5 - lecture de la valeur associée à une propriété adaptée à spécifier le nœud du réseau de communication adapté à exécuter le traitement ;
- lecture de la valeur d'une propriété adaptée à spécifier si le traitement est obligatoire ou optionnel ; et
- vérification si le traitement est supporté par l'ordinateur client du
- 10 réseau de communication lorsque le traitement est obligatoire et doit être exécuté par l'ordinateur client du réseau de communication.

Il est ainsi possible à l'ordinateur client de tester s'il est adapté à inter-agir avec un service proposé par un ordinateur serveur.

- Selon un troisième aspect de l'invention, elle concerne un procédé
- 15 de validation d'un message reçu par un ordinateur intermédiaire du réseau de communication, à partir d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML du message échangé lors de l'exécution de ce service sur le réseau de communication.

- 20 Ce procédé de validation comprend les étapes suivantes :

- extraction d'un traitement dans le message reçu ;
- acquisition d'au moins une valeur impérative associée à une propriété du traitement ; et
- vérification si la valeur impérative est incluse dans une liste de
- 25 valeurs attribuables à une propriété supportée par la fonctionnalité décrite dans le document de description d'un service.

- Il est ainsi possible à partir d'un message et des valeurs des propriétés associées à chaque traitement inclus dans le message, de vérifier s'il correspond à la description du traitement tel que décrit dans la fonctionnalité du
- 30 document de description d'un service.

De préférence, lors de cette validation du message par un ordinateur intermédiaire, le procédé comprend une étape de lecture de la valeur associée

à une propriété adaptée à spécifier si le traitement est exécuté avant ou après l'envoi du message et une étape d'exécution de ce traitement lorsque la valeur est adaptée à spécifier que le traitement doit être exécuté avant l'envoi du message.

5 La présente invention concerne également un dispositif de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication.

10 Ce dispositif de proposition d'un service comprend des moyens d'envoi d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution de ce service sur le réseau de communication.

15 Elle concerne également un dispositif de test d'accès à un service par un ordinateur client d'un réseau de communication, à partir d'un document de description d'un service, comprenant :

- des moyens d'extraction d'une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication ;
- 20 - des moyens de lecture de la valeur associée à une propriété adaptée à spécifier le terminal du réseau de communication adapté à exécuter le traitement ;
- des moyens de lecture de la valeur d'une propriété adaptée à spécifier si le traitement est obligatoire ou optionnel ; et
- 25 - des moyens de vérification adaptés à vérifier si le traitement est supporté par l'ordinateur client du réseau de communication lorsque ce traitement est obligatoire et doit être exécuté par l'ordinateur client du réseau de communication.

30 La présente invention concerne également un dispositif de validation d'un message reçu par un ordinateur intermédiaire du réseau de communication, à partir d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou

du post-traitement de données en format XML du message échangé lors de l'exécution d'un service sur le réseau de communication.

Ce dispositif de validation comprend :

- des moyens d'extraction d'un traitement dans un message reçu ;
- 5 - des moyens d'acquisition d'au moins une valeur impérative associée à une propriété du traitement ; et
- des moyens de vérification adaptés à vérifier si la valeur impérative est incluse dans une liste de valeurs attribuables à une propriété supportée par la fonctionnalité décrite dans le document de description d'un
- 10 service.

Ces dispositifs de proposition d'un service, de test d'accès et de validation d'un message reçu présentent des caractéristiques et avantages analogues à ceux des procédés qu'ils mettent en œuvre.

Par ailleurs, la présente invention concerne un ordinateur comportant

15 un dispositif de proposition d'un service ou un dispositif de test d'accès ou un dispositif de validation d'un message conformes à l'invention.

Elle vise tout particulièrement un ordinateur serveur d'un réseau de communication comprenant des moyens adaptés à mettre en œuvre le procédé de proposition d'un service conforme à l'invention.

20 Corrélativement, elle vise un ordinateur client d'un réseau de communication comprenant des moyens adaptés à mettre en œuvre le procédé de test d'accès conforme à l'invention.

La présente invention vise également un ordinateur d'un réseau de communication comprenant des moyens adaptés à mettre en œuvre le procédé

25 de validation d'un message conforme à l'invention.

Plus généralement, la présente invention concerne un réseau de communication comprenant des moyens adaptés à mettre en œuvre le procédé de proposition d'un service et/ou le procédé de test d'accès et/ou le procédé de validation d'un message conformes à l'invention.

30 Elle vise par ailleurs un moyen de stockage d'informations, éventuellement totalement amovible, lisible par un système informatique, comprenant des instructions pour un programme informatique adaptées à

mettre en œuvre le procédé de proposition d'un service et/ou le procédé de test d'accès et/ou le procédé de validation d'un message conforme à l'invention, lorsque ce programme est chargé et exécuté par le système informatique.

Enfin, elle concerne un programme d'ordinateur lisible par un microprocesseur comprenant des portions de code logiciel adaptées à mettre en œuvre le procédé de proposition d'un service et/ou le procédé de test d'accès et/ou le procédé de validation d'un message conformes à l'invention, lorsque ce programme d'ordinateur est chargé et exécuté par le microprocesseur.

Les ordinateurs, l'ordinateur serveur, l'ordinateur client, le réseau de communication, le moyen de stockage d'informations et programme d'ordinateur présentent des caractéristiques et avantages analogues aux procédés qu'ils mettent en œuvre.

D'autres particularités et avantages de l'invention apparaîtront encore dans la description ci-après.

Aux dessins annexés, donnés à titre d'exemples non limitatifs :

- la figure 1 est un algorithme illustrant un procédé de proposition d'un service conforme à l'invention ;
- la figure 2 est un algorithme illustrant un procédé de test d'accès à un service conforme à un mode de réalisation de l'invention ;
- la figure 3 est un algorithme illustrant un procédé de validation d'un message reçu par un ordinateur conforme à un mode de réalisation de l'invention ;
- la figure 4 est un algorithme illustrant l'exécution d'un message incluant un traitement ; et
- la figure 5 est un schéma bloc illustrant un dispositif adapté à mettre en œuvre l'invention.

On va décrire tout d'abord en référence à la figure 1 un procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication.

Dans un réseau d'ordinateur tel qu'Internet, des serveurs fournissent des données sous forme de documents à des ordinateurs clients.

Très fréquemment, ces ordinateurs serveurs fournissent également des services, appelés services Web, permettant à un ordinateur client d'envoyer des données au serveur qui les traite et retourne un résultat.

5 En pratique, lorsqu'un ordinateur client d'un réseau de communication souhaite utiliser les services fournis par un ordinateur serveur, il envoie un message afin de requérir une description des services fournis par cet ordinateur serveur.

10 A la réception E10 d'une telle requête de description, le procédé de proposition de services comprend une étape d'envoi E11 d'un document de description d'un service.

Dans la suite de la description, et de manière non limitative, on considérera que ce document de description d'un service utilise le langage WSDL (acronyme du terme anglais "Web Service Description Language"), qui est un langage permettant de décrire des services Web.

15 Ce langage WSDL est un langage XML perfectionné, permettant à l'aide de balises XML de décrire un service Web.

Un document WSDL contient principalement deux parties. Une première partie, appelée partie abstraite, décrit de manière abstraite les messages qui sont échangés entre ordinateurs du réseau de communication
20 lors de la mise en œuvre du service. Une seconde partie est adaptée à comporter les informations concrètes, définissant la transmission des messages sur le réseau de communication.

La première partie est elle-même divisée en trois sous-parties.

25 La première sous-partie de cette première partie contient des déclarations de types, permettant de décrire la structure abstraite des messages. Ces types sont ensuite référencés dans la deuxième sous-partie de la première partie du document WSDL.

La déclaration de ces types est généralement réalisée au moyen de balises `<types>...</types>`.

30 Cette déclaration de types est bien connue dans les documents écrits en langage XML et il n'est pas nécessaire pour la compréhension de l'invention de décrire plus avant les types utilisés.

La seconde sous-partie de la première partie du document contient des déclarations de messages élémentaires.

Elle définit ainsi les messages qui seront échangés entre les ordinateurs lors de la mise en œuvre du service, sans en décrire précisément le contenu ou l'enchaînement.

Elle consiste à définir de manière abstraite et de donner le type d'une donnée transmise (entrée ou sortie).

A titre d'exemple, et uniquement pour faciliter la compréhension de la présente invention, les messages suivants peuvent être définis, sans lien entre eux :

a) un premier message permet de transmettre le nom d'une action :

```
<message name="GetStockQuoteInput">  
  <part name="Name" type="string"/>  
</message>
```

b) un second message permet de transmettre le prix associé à une action :

```
<message name="GetStockQuoteOutput">  
  <part name="Price" type="float"/>  
</message>
```

Une troisième sous-partie de la première partie du document WSDL permet de regrouper les messages élémentaires ainsi définis dans les deux premières sous-parties en opérations logiques.

Une opération peut être considérée comme un service élémentaire, ce dernier étant lui-même implanté par un ou plusieurs messages.

Elle est ainsi définie par ses entrées et sorties, c'est-à-dire par les messages qu'elle échange.

Par exemple, une opération retournant la valeur d'une action lors de la réception du nom de cette action peut être définie comme suit :

```
<operation name="GetStockQuote">  
  <input message="tns:GetStockQuoteInput"/>  
  <output message="tns:GetStockQuoteOutput"/>  
</operation>
```

Bien évidemment des formes plus élaborées d'opération, constitué d'un ensemble complexe d'échanges de messages, pourraient être décrites dans ce langage.

Cette première partie du document WSDL permet ainsi de définir le type, le contenu et l'ordonnancement des messages échangés entre ordinateurs du réseau lors la mise en œuvre d'un service proposé par un ordinateur serveur.

5 Le document WSDL comprend en outre, comme expliqué précédemment, une seconde partie concrète qui permet de préciser quel protocole est effectivement utilisé pour transmettre les messages et quel encodage ou format de représentation est mis en œuvre pour représenter ces données sous une forme adaptée au réseau de communication.

10 Cette seconde partie correspond ainsi à un couplage (binding) consistant à spécifier un protocole concret et un format de données pour une opération définie dans la première partie du document WSDL.

 Conformément à l'invention, le document de description d'un service comprend la description d'une fonctionnalité mise en œuvre lors d'un pré-
15 traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication.

 Cette fonctionnalité permet ainsi de définir des traitements à effectuer sur un message.

 Il s'agit en particulier de traitements qui utilisent ou produisent des
20 données en format XML qui sont définies dans la partie abstraite du document de description d'un service WSDL.

 On notera cependant qu'un traitement peut combiner plusieurs traitements.

 Cette description de fonctionnalité consiste à définir une
25 fonctionnalité par une liste de propriétés supportées par cette fonctionnalité.

 Il s'agit alors de donner une description, par exemple dans un langage de balisage du type XML, de la fonctionnalité en décrivant les propriétés associées à cette fonctionnalité. Ces éléments ou fonctionnalités ainsi définis, ils peuvent être utilisés pour être inclus dans une description de
30 service du type d'un document WSDL.

On obtient ainsi un document de description d'un service écrit dans un langage WSDL perfectionné, permettant de décrire directement dans ce document les traitements à effectuer sur un message.

Les propriétés de cette fonctionnalité permettent de déterminer en particulier sur quel message ou sur quel ensemble de message le traitement s'applique ; le moment auquel le traitement est effectué, et notamment à l'envoi ou à la réception du message par un ordinateur ; quelles données du message le traitement utilise ou génère et le type de données ainsi générées.

Lorsque le traitement est mis en œuvre via un langage de script, il est nécessaire de définir également les scripts possibles pour l'exécution de ce traitement.

En outre, il est avantageux qu'une propriété permette de spécifier si le traitement à effectuer est obligatoire, ou optionnel, ou encore négociable.

On va donner ci-après un premier exemple de description en langage XML d'une telle fonctionnalité de traitement.

La description XML d'une fonctionnalité est ainsi encapsulée dans un élément XML. On considère ici de manière non limitative que la fonctionnalité est du type d'un élément ("feature") tel que défini par la norme SOAP.

Cet élément comporte un attribut obligatoire "name" permettant de donner un nom au traitement et de référencer ultérieurement la description de cette fonctionnalité.

En outre, la liste des propriétés supportées par cette fonctionnalité est insérée à l'intérieur de l'élément. Chaque propriété est définie par un élément que l'on nomme "property". Cet élément a un attribut obligatoire "name". Pour chaque propriété, la description de cette fonctionnalité comprend une liste de valeurs attribuables à cette propriété.

Ainsi, à titre d'exemple, la fonctionnalité "process feature" peut être décrite de la façon suivante, sous la forme de schéma XML :

```
<feature name="ProcessFeature">
  <property name="process" type="ProcessType"/>
  <property name="value" type="ValueType"/>
  <property name="data" type="DataType"/>
  <property name="rule" type="RuleType"/>
</feature>
```

```

<xs:complexType name="ProcessType">
  <xs:element name="process">
    <xs:complexType>
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="side">
        <xs:simpleType>
          <xs:restriction base="xs:NSTOKEN">
            <xs:enumeration value="sender"/>
            <xs:enumeration value="receiver"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="order" type="xs:integer"
use="optional"/>
      <xs:attribute name="msg" type="xs:NCNAME"/>
    </xs:complexType>
  </xs:element>
</xs:complexType>

<xs:complexType name="ValueType">
  <xs:element name="type" maxOccurs="unbounded">
    <xs:complexType>
      <xs:element name="value" minOccurs="0">
        <xs:complexType>
          <xs:choice>
            <xs:any processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="value">
              <xs:complexType>
                <xs:any processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
              </xs:complexType>
            </xs:element>
          </xs:choice>
        </xs:complexType>
      </xs:element>
      <xs:anyAttribute processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:complexType>
  </xs:element>
</xs:complexType>

<xs:complexType name="DataType">
  <xs:element name="data">
    <xs:complexType>
      <xs:element name="input" minOccurs="0">
        <xs:complexType>
          <xs:attribute name="ref" use="optional"/>
          <xs:attribute name="type" use="optional"/>
          <xs:anyAttribute processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
          <xs:any processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:complexType>

```



```

        </xs:complexType>
      </xs:element>
      <xs:element name="output" minOccurs="0">
        <xs:complexType>
          <xs:attribute name="ref" type="xs:string"
            use="optional"/>
          <xs:attribute name="type" type="xs:string"
            use="optional"/>
          <xs:anyAttribute processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
          <xs:any processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:complexType>

<xs:complexType name="RuleType">
  <xs:element name="rule">
    <xs:simpleContent>
      <xs:attribute name="value" type="xs:boolean"
        use="required">
        <xs:simpleContent>
          <xs:restriction base="xs:string">
            <xs:enumeration value="mandatory"/>
            <xs:enumeration value="optional"/>
          </xs:restriction>
        </xs:simpleContent>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:element>
</xs:complexType>

```

Ainsi, dans cet exemple, la fonctionnalité "process feature" a une propriété, nommée "process", qui permet de définir le message cible et l'entité de traitement, c'est-à-dire le nœud du réseau de communication adapté à exécuter le traitement.

5 Elle comporte également une propriété qui a pour nom "value" et qui permet de définir le type et la valeur du traitement.

La fonctionnalité comporte également une propriété ayant pour nom "data" qui définit les données en entrée et en sortie du traitement, et notamment le type de données générées.

10 En outre, une propriété "rule" permet de définir si le traitement à effectuer est obligatoire ou optionnel.

La description de chaque propriété est ainsi décrite à l'aide d'un langage de schéma, tel que XML-schéma, de telle sorte que la description de chaque propriété est insérée par référence dans la description de la fonctionnalité.

5 Ainsi, dans l'exemple précédent, la propriété "process" définit le message sur lequel s'applique le traitement sous la forme d'un attribut ayant pour nom "msg". Cet attribut peut définir un message particulier, ou encore un ensemble de messages, par exemple l'ensemble des messages envoyés par un ordinateur serveur.

10 Cette propriété "process" comporte également un attribut "side" permettant de définir si le traitement est effectué à l'envoi, par un émetteur, ou à la réception, par un récepteur du message.

Ainsi, les valeurs qui peuvent être pris par cet attribut "side" sont typiquement "sender" ou "receiver" (émetteur ou récepteur).

15 Une seconde propriété "value" permet de définir les valeurs qui peuvent être prises par un script lors du traitement. Cette propriété permet de définir le type de script utilisé via un élément "type" comprenant un attribut "name". A l'intérieur de cet élément "type", il est possible d'ajouter des contraintes, comme par exemple la possibilité d'utiliser certaines parties du
20 langage.

Il est également possible de proposer le choix entre plusieurs scripts. Enfin, un même traitement peut supporter plusieurs types de script ; il suffit alors d'ajouter dans la description autant d'éléments "type" que de types de scripts.

25 Une propriété "data" permet en outre de définir les données utilisées via un élément "input" et les données produites via un élément "output". Ces éléments peuvent contenir un attribut "ref" qui détermine la partie du message. Dans l'exemple précédent, le type des données est spécifié au moins en entrée ou en sortie du traitement.

30 Il est ainsi possible de déterminer si les données en entrée sont valides, et après exécution du script, si les données en sortie sont valides.

Ce type de données peut être spécifié indirectement via une définition abstraite du message dans un document WSDL ou bien directement par un attribut "type".

Enfin, une propriété "rule" permet de définir si le traitement à effectuer est obligatoire ou optionnel.

En l'absence de cette propriété, le traitement est obligatoire.

Bien que l'on ait décrit ci-dessus un exemple de fonctionnalité utilisant le protocole SOAP, permettant le traitement de données abstraites via une feature SOAP, n'importe quel type de fonctionnalité, s'appliquant alors à tout autre protocole, pourrait être décrit de la même manière au niveau abstrait d'un document de description d'un service WSDL.

Ainsi, on peut implémenter une fonctionnalité de traitement au niveau abstrait du document de description d'un service WSDL, en ajoutant une donnée XML au message, traduisant un traitement à effectuer sur des données.

Cette donnée XML est ensuite encodée (ou sérialisée) au niveau de la seconde partie concrète du document de description d'un service WSDL comme tout autre partie de message abstrait.

Ainsi, cette fonctionnalité est offerte à tout type de protocole, et par exemple au protocole SOAP, au protocole HTTP ou encore au protocole BEEP dont une description peut être consultée à l'adresse suivante : <http://www.ietf.org/rfc/rfc3080.txt>

On définit alors dans la description de cette fonctionnalité un encodage par défaut dépendant du protocole, tel que par exemple :

- pour HTTP, une partie message mime ;
- pour SOAP, un en-tête particulier ; et
- pour BEEP, un élément XML particulier.

Ainsi il n'est pas nécessaire de spécifier des données concrètes dans le document. Cependant si l'on souhaite spécifier des informations concrètes dans le document WSDL, on utilise une référence à la propriété concernée.

Un exemple d'une telle référence est donnée ci-après :

```

<portType name="fooPT">
  <operation name="fooLOP">
    <input message="fooMsgIn" name="in"/>
    <output message="fooMsgOut" name="out"/>
    <feature ref="ProcessFeature" mustUse="false">
      <property name="process">
        <process name="XSLT_IN" side="receiver"
msg="name:fooMsgIn"/>
      </property>
      <property name="http://xslt-feature/value">
        <value>
          <type name="XSLT"/>
        </value>
      </property>
      <property name="data">
        <data>
          <output ref="part:*/>
        </data>
      </property>
    </feature>
  </portType>
<binding name="fooSoap" type="fooPt">
  <soap:binding transport="http://example.com/soap/http"/>
  <operation name="fooLOP">
    <input>
      <soap:body part="xml-data1" .../>
      <soap:header part="xml-data2" .../>
      <soap:header part="http://xslt-feature/value" .../>
    </input>
    <output/>
  </operation>
</binding>
<binding name="fooHttp" type="fooPt">
  <http:binding transport="http://example.com/http"/>
  <operation name="fooLOP">
    <input>
      <mime:multipartRelated>
        <mime:part>
          <mime:content
part="xml-data1" use="literal"/>
        </mime:part>
        <mime:part>
          <mime:content part="xml-data2" use="literal"/>
        </mime:part>
        <mime:part>
          <mime:content
part="http://xslt-feature/value"
type="text/my-op-script" />
        </mime:part>
      </mime:multipartRelated>
    </input>
    <output/>
  </operation>

```

</binding> .

Ainsi, il est possible d'encoder la valeur d'une propriété abstraite au moyen d'un mécanisme de référence identique à celui utilisé pour encoder des données abstraites des messages.

En outre, un encodage par défaut peut être défini dans un format
5 compréhensible par une machine pour différents protocoles.

On revient à présent au premier exemple de réalisation dans lequel la fonctionnalité est du type d'un élément "feature" tel que défini par la norme SOAP

Une fois les propriétés de la fonctionnalité définies, on peut décrire
10 un service implémentant cette fonctionnalité au travers d'un document de description d'un service WSDL.

De préférence, on intègre la description de cette fonctionnalité, via un élément "feature" placé dans la partie abstraite du document de description d'un service, dès lors que cette fonctionnalité est adaptée à produire ou utiliser
15 des données en format XML définies dans cette même partie abstraite.

A l'intérieur de l'élément "feature", les valeurs prises par les différentes propriétés sont spécifiées, parmi la liste de valeurs attribuables à chaque propriété. En d'autres termes, la définition de la fonctionnalité donne un ensemble de valeurs que peut prendre chaque propriété, et le document de
20 description d'un service WSDL est adapté à définir un sous-ensemble de ces valeurs de propriété, c'est-à-dire à ajouter des contraintes sur les valeurs que peuvent prendre ces propriétés.

On donne un exemple ci-dessous d'une utilisation de la fonctionnalité décrite précédemment telle qu'elle peut être incorporée dans un document de
25 description d'un service du type WSDL.

```
<portType name="fooPT">
  <operation name="fooLOP">
    <input message="fooMsgIn" name="in" />
    <output message="fooMsgOut" name="out" />
    <feature ref="ProcessFeature" mustUse="false">
      <property name="process">
        <process name="XSL_IN" side="receiver"
msg="name:fooMsgIn"/>
      </property>
```

```

    <property name="value">
      <value>
        <type name="XSL"/>
      </value>
    </property>
    <property name="data">
      <data>
        <output ref="part:*/>
      </data>
    </property>
  </feature>
  <feature ref="ProcessFeature" mustUse="true">
    <property name="process">
      <process name="JS_OUT" side="receiver"
msg="name:foolMsgOut"/>
    </property>
    <property name="value">
      <value>
        <type name="JavaScript+DOM">
          <!--might contain constraints on the process-->
          <platform>neutral</platform>
          <access>internet</access>
        </type>
      </value>
    </property>
    <property name="data">
      <data>
        <output ref="part:part1"/>
      </data>
    </property>
  </feature>
</portType>
<binding name="fooSoap" type="fooPt">
  <soap:binding transport="http://example.com/http"/>
  <operation name="foolOP">
    <input/>
    <output/>
    <feature ref="ProcessFeature" mustUse="false">
      <property name="process">
        <process name="XSL_OUT" side="sender"
msg="name:foolMsgOut"/>
      </property>
      <property name="value">
        <value>
          <type name="XSL"/>
        </value>
      </property>
      <property name="output">
        <data>
          <input ref="part:*/>
        </data>
      </property>
    </feature>

```

```

</operation>
<feature ref="ProcessFeature" mustUse="false">
  <property name="process">
    <process name="PERL_IN" side="receiver"
msg="type:inbound"/>
  </property>
  <property name="value">
    <value>
      <type name="PERL">
        <library usable="DOM"/>
        <library unusable="HTTP"/>
      </type>
    </value>
  </property>
</feature>
</binding>

```

Ainsi, dans l'exemple précédent quatre utilisations de la fonctionnalité sont illustrées :

1) Le premier traitement XSL_IN propose à un client d'inclure dans sa requête un script XSL (<type name="XSL"/>) qui va être exécuté au niveau du service (side="receiver") et produira les données telles que demandées par le service. Ce traitement est optionnel (mustUse="false"), et s'applique sur une ou plusieurs parties (<output ref="part:*/>) du message 'foo1MsgIn' (msg="name:foo1MsgIn").

10 Ainsi, il est possible pour un nœud intermédiaire d'un réseau de communication d'ajouter un script XSL à un message et d'envoyer celui-ci à un récepteur final qui exécutera le script, puis traitera la requête.

On diminue ainsi la charge de travail d'un nœud intermédiaire qui n'a pas besoin d'implémenter le langage de script, ici le script XSL.

15 2) Le deuxième traitement JS_OUT oblige un client à supporter les scripts JavaScript dans les messages de retour ;

3) Le troisième traitement XSLT_OUT propose au client de fournir un script XSLT qui sera appliqué sur le message de retour ; et

20 4) Le quatrième traitement PERL_IN propose au client d'ajouter des scripts PERL dans tous les messages reçus par le service (msg="type:inbound"). Le traitement peut utiliser les fonctionnalités DOM

(`<library usable="DOM"/>`) de PERL mais ne doit pas utiliser les fonctionnalités HTTP (`<library unusable="HTTP"/>`).

Il est ainsi possible de décrire des traitements appliqués à différents messages échangés lors de la mise en œuvre d'un service Web sur un réseau de communication.

On va décrire à présent en référence à la figure 2 un procédé de test d'accès à un service à partir d'un document de description d'un service tel que décrit précédemment.

Ce procédé de test d'accès permet à un ordinateur client de vérifier s'il est capable d'interagir avec un service proposé par un ordinateur serveur du réseau de communication, à la réception d'un document de description d'un service du type WSDL tel que décrit précédemment.

Pour vérifier cet accès, l'ordinateur client est adapté à tester différentes choses, et en particulier les protocoles utilisables pour exécuter le service.

En particulier, et on se limitera dans la description ci-après à ce point, l'ordinateur client est adapté à vérifier s'il est adapté à implémenter la fonctionnalité de description d'un pré-traitement ou d'un post-traitement des données, et s'il est en mesure d'effectuer chaque traitement demandé.

En particulier, l'ordinateur client vérifiera le type de script proposé pour la mise en œuvre du traitement, via la description de la propriété "value" décrite précédemment.

Une première étape de lecture E21 est mise en œuvre pour la lecture du document WSDL adapté à décrire le service.

Une étape de sélection E22 est adaptée à sélectionner une opération décrite dans le document WSDL, c'est-à-dire une suite de messages échangés lors de l'exécution d'un service sur le réseau de communication.

Pour cette opération, une étape de test E23 est adaptée à vérifier s'il existe un traitement décrit associé à cette opération.

En pratique, une étape d'extraction de la description de la fonctionnalité est mise en œuvre.

Si aucun traitement n'est décrit, une réponse est adressée dans une étape de réponse E27 afin d'indiquer que l'accès à l'opération est possible pour l'ordinateur client.

5 Sinon, une étape de test E24 permet de vérifier si le traitement est obligatoire et doit être exécuté par l'ordinateur client du réseau de communication.

En pratique, une étape de lecture permet de lire une valeur associée à la propriété "process" adaptée à spécifier le nœud du réseau de communication devant exécuter le traitement.

10 Une étape de lecture de la valeur de la propriété "rule" permettant de spécifier si le traitement est obligatoire ou optionnel, est également mise en œuvre lors de cette étape de test E24.

15 Lorsque ce traitement est obligatoire et doit ainsi être exécuté par l'ordinateur client du réseau de communication, une étape de vérification E25 permet de vérifier si le traitement est supporté par l'ordinateur client du réseau de communication.

En particulier, l'ordinateur client doit vérifier si le type de script proposé via la description de la propriété "value" est bien implémenté par l'ordinateur client.

20 Dans l'affirmative, si ce traitement est supporté par l'ordinateur client, on réitère pour les autres traitements associés à l'opération les étapes de test E23 à E25.

25 Si à l'issue de l'étape de vérification E25, le traitement obligatoire et exécutable par l'ordinateur n'est pas supporté par celui-ci, un message déclarant l'accès à l'opération impossible est adressé dans une étape de réponse E26 à l'ordinateur client.

30 A contrario, si tous les traitements sont supportés par l'ordinateur client, à l'issue de l'étape de test E23, lorsque tous les traitements ont été analysés, une réponse est adressée à l'ordinateur client dans une étape de réponse E27 indiquant que l'accès à l'opération est possible.

On va décrire à présent en référence à la figure 3 un procédé de validation d'un message reçu par un ordinateur du réseau de communication.

Ce procédé de validation d'un message permet à un nœud intermédiaire du réseau, situé entre un émetteur et un récepteur d'un message, de valider le message comprenant des traitements. Cette validation du message est réalisée grâce à la connaissance d'un document de description d'un service WSDL tel que décrit précédemment.

Ce nœud intermédiaire du réseau de communication peut éventuellement effectuer des traitements. S'il n'effectue pas de traitement, il est adapté à vérifier que l'ordinateur récepteur est capable d'effectuer les traitements sur le message envoyé. Si ni le récepteur et ni le nœud intermédiaire ne sont capables d'effectuer le traitement, l'ordinateur intermédiaire peut éventuellement diriger le message vers un autre nœud du réseau de communication adapté à mettre en œuvre le traitement.

Si cet ordinateur intermédiaire effectue des traitements, il vérifiera si les données produites à l'issue de ce traitement sont bien valides vis-à-vis du type de données affecté aux données tel que décrit dans le document de description WSDL conforme à l'invention.

En pratique, une étape d'acquisition E30 du message est mise en œuvre au niveau du nœud intermédiaire.

On identifie ensuite dans une étape d'extraction E31 la description du service associé grâce à la lecture d'un document de description de service WSDL conforme à l'invention.

On vérifie dans une étape de test E32 s'il existe des traitements à effectuer. Dans la négative, une étape d'utilisation E33 est adaptée à utiliser le message dès lors qu'il n'y a pas de traitement à réaliser. Sinon, une étape d'extraction E34 permet d'extraire un traitement dans le message reçu.

Une étape de validation E35 est ensuite mise en œuvre afin de valider le traitement par rapport à la description faite dans le document de description d'un service WSDL.

En pratique on acquiert au moins une valeur impérative associée à une propriété du traitement telle que définie dans le message et on vérifie dans une étape de vérification si cette valeur impérative est bien incluse dans une

liste de valeurs attribuables à la propriété supportée par la fonctionnalité telle que décrite dans le document de description d'un service WSDL.

Si à l'issue de cette étape de validation E35, le traitement n'est pas validé, une étape d'envoi E40 est adaptée à envoyer un message d'erreur à l'ordinateur client.

Sinon, si le traitement est validé à l'issue de l'étape de validation E35, une étape de test E36 est mise en œuvre afin de vérifier s'il s'agit ou non d'un pré-traitement.

En pratique, lors de cette étape de test E36, la valeur associée à une propriété adaptée à spécifier si le traitement est exécuté avant ou après l'envoi du message est lue.

Si cette valeur est adaptée à spécifier que le traitement doit être exécuté avant l'envoi du message, une étape d'exécution E37 est alors mise en œuvre afin d'exécuter ce pré-traitement.

Sinon, si à l'issue de l'étape de test E36, la valeur de la propriété est adaptée à spécifier que le traitement doit être exécuté après l'envoi du message, on en déduit qu'il s'agit d'un post-traitement et les étapes E32 et suivantes sont répétées pour un nouveau traitement, s'il en existe un dans le message.

Après l'étape d'exécution E37 d'un pré-traitement, une étape de validation E38 est mise en œuvre afin de valider le résultat obtenu par le pré-traitement par rapport au type de données spécifié dans le document de description WSDL.

On teste dans une étape de test E39 si cette validation est réussie ou non. Dans la négative, l'étape d'envoi E40 d'un message d'erreur est mise en œuvre.

Sinon, l'ensemble des étapes E32 et suivantes est répété pour un autre traitement incorporé dans le message.

On va décrire à présent en référence à la figure 4 un procédé de traitement d'un message comportant des traitements.

Chaque message possède un contexte qui lui est associé. Ce contexte possède des informations de traitement non-incluses dans le

message. Par exemple, ces informations peuvent venir de messages précédents contenant des informations de traitement concernant le message.

Une fois le message et son contexte acquis, on énumère les traitements. Chaque traitement est affecté au contexte (créé si nécessaire) du message cible, ce message cible pouvant être différent du message contenant les informations de traitement.

Ce contexte représente ainsi une liste ordonnée de traitements.

Lorsque les traitements inclus dans le message ont été énumérés, on effectue l'ensemble des traitements inclus dans le contexte, dans l'ordre spécifié par le contexte. Après avoir exécuté ce traitement, on valide les résultats en fonction du type de résultat attendu. Dans le cas d'une validation positive, on ajoute ces résultats au message. Lorsqu'il n'y a plus de traitement à effectuer, le message peut être utilisé, c'est-à-dire envoyé ou traité comme un appel de fonction à distance.

En pratique, une première étape d'acquisition E41 est mise en œuvre afin de réceptionner le message. Une étape d'acquisition E42 du contexte du message est également effectuée. On obtient ainsi la description de l'échange associé au message. On vérifie ensuite dans une étape de test E43 s'il existe des traitements à effectuer sur le message.

Dans l'affirmative, on extrait et on analyse l'ensemble de ces traitements. Ainsi, une étape d'acquisition E44 est mise en œuvre afin d'obtenir le traitement suivant incorporé dans le message. Dans une étape d'extraction E45, le message cible est extrait du traitement et dans une étape d'ajout E46, le traitement est ajouté au contexte du message cible.

Ainsi, ces différentes étapes E44 à E46 permettent d'entreposer dans le contexte du message cible l'ensemble des traitements à ajouter sur ce message.

Lorsqu'à l'issue de l'étape de test E43, il n'y a plus de traitement à effectuer sur le message, on vérifie dans une étape de test E47 s'il existe des traitements à faire, tels qu'ils ont été incorporés dans le contexte.

S'il s'agit d'un traitement immédiat des données du message, on acquiert dans une étape d'acquisition E48 le traitement et on exécute celui-ci.

Une étape d'extraction E49 permet d'extraire les résultats du traitement et de valider ces résultats en fonction des données types attendues à l'issue du traitement.

5 Lorsque ces résultats sont validés, une étape d'intégration E50 permet d'intégrer le résultat du traitement au message.

On réitère les étapes de traitement E48 à E50 tant qu'il existe des traitements à effectuer dans le contexte.

10 A l'issue de l'étape E47, lorsqu'il n'existe plus de traitement à réaliser au niveau du nœud considéré, les éventuels autres traitements devant être effectués après envoi du message par un autre ordinateur du réseau, ces traitements sont encodés comme information de la fonctionnalité et sont ajoutés au message.

15 Lorsqu'il n'y a plus de traitement, le message est utilisé dans l'étape d'utilisation E51, c'est-à-dire que le message est envoyé au nœud suivant dans le réseau de communication.

20 On décrit ci-après deux exemples d'encodage des informations d'une fonctionnalité qui sont introduites dans un message. Ainsi, on encode un traitement à effectuer dans un message, lors d'une interaction entre un client et un serveur. Les valeurs des propriétés sont fixées et la fonctionnalité encode ces informations dans le message.

Ces informations sont mises dans l'en-tête (en anglais "header") du message.

```
<Soap:Envelope
  xmlns:Soap="http://schemas.xmlsoap.org/soap/envelope/"
  <Soap:Header>
    <sf:feature name="XSLT_IN" id="1" (msg="self"
      side="receiver")
  </Soap:Header>
  xmlns:sf="http://examples.com/soap/ProcessFeature"
  Soap:role="http://www.w3.org/2002/06/soap-
  envelope/role/next"
  Soap:mustUnderstand="true">
    <sf:data>
      <input ref="part:part1"/>
      <output ref="part:part1"/>
    </sf:data>
    <sf:processing type="XSLT">
      <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/.../Transform">
```

```

        <xsl:template match="/">
            ...
        </xsl:template>
    </xsl:stylesheet>
</sf:processing>
</sf:feature>
</Soap:Header>
<Soap:Body>
    <part1 sf:name="xsltin" sf:id="1"/>
    <part2>
        ...
    </part2>
</Soap:Body>
</Soap:Envelope>

```

Dans cet exemple précédent, le message comporte un script à exécuter sur ce message à la réception. Ce script est du type XSL et va utiliser des données contenues dans la partie "part1" du message et les remplacer par le résultat du script.

- 5 Le schéma des données en entrée n'est pas donné, mais il pourrait être ajouté via un attribut "type" de l'élément "input". Le schéma des données en sortie n'est pas donné explicitement mais il doit correspondre au schéma décrit dans le document de description d'un service WSDL du service recevant le message.

```

<Soap:Envelope
  xmlns:Soap="http://schemas.xmlsoap.org/soap/envelope/">
  <Soap:Header>
    <sf:feature
      xmlns:sf="http://examples.com/soap/ProcessFeature"
      name="XSLT_OUT" id="1" msg="foolMsgOut"
      side="sender"
      Soap:role="http://www.w3.org/2002/06/soap-
        envelope/role/next"
      Soap:mustUnderstand="true">
        <sf:data>
          <input ref="part:part1"/>
          <output ref="part:part1"/>
        </sf:data>
        <sf:processing type="XSLT">
          <xsl:stylesheet version="1.0"
            xmlns:xsl="http://www.w3.org/.../Transform">
            <xsl:template match="/">
              ...
            </xsl:template>
          </xsl:stylesheet>
        </sf:processing>
      </sf:feature>
    </Soap:Header>

```

```
<Soap:Body>
  <part1 sf:name="xsltin " sf:id="1"/>
  <part2>
    ...
  </part2>
</Soap:Body>
</Soap:Envelope>
```

Dans ce second exemple, le message comporte un script à exécuter sur un message "foo1MsgOut" avant l'envoi du message. Le script est également de type XSL et utilise les données contenues dans la partie "part1" du message. Le traitement est adapté à remplacer ces données par le résultat du script.

Dans cet exemple, le nœud qui reçoit le message va stocker l'information et l'utiliser correctement : s'il émet le message "foo1MsgOut", il appliquera le script à ce message avant de l'envoyer. Sinon, il enverra les informations de traitement au nœud qui émettra le message "foo1MsgOut".

La présente invention permet ainsi de définir un moyen de description d'un service qui permet de décrire les pré-traitements et les post-traitements que le service peut effectuer.

Afin de mettre en œuvre le procédé de proposition d'un service tel que décrit précédemment en référence à la figure 1, un dispositif de proposition d'un service comprend essentiellement des moyens d'envoi d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication.

De même, un dispositif de test d'accès par un ordinateur client d'un réseau de communication comprend essentiellement des moyens d'extraction d'une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML d'un message échangé lors de l'exécution d'un service sur le réseau de communication, des moyens de lecture de la valeur associée à une propriété adaptée à spécifier le nœud du réseau de communication adapté à exécuter le traitement et de la valeur d'une propriété adaptée à spécifier si le traitement est obligatoire ou optionnel, et des moyens de vérification si le traitement est supporté par l'ordinateur client du réseau de

communication lorsque le traitement est obligatoire et doit être exécuté par l'ordinateur client du réseau de communication.

Enfin, un dispositif de validation d'un message par un ordinateur intermédiaire d'un réseau de communication comprend des moyens d'extraction
5 d'un traitement dans le message reçu, des moyens d'acquisition d'au moins une valeur impérative associée à une propriété du traitement et des moyens de vérification adaptés à vérifier si la valeur impérative est incluse dans une liste de valeurs attribuables à une propriété supportée par la fonctionnalité décrite dans le document de description d'un service.

10 Ces dispositifs de proposition d'un service, de test d'accès à un service et de validation d'un message peuvent être incorporés dans un ordinateur tel qu'illustré à la figure 5.

En particulier, le dispositif de proposition d'un service est incorporé dans un ordinateur serveur S d'un réseau de communication alors que le
15 dispositif de test d'accès à un service est incorporé dans un ordinateur client C d'un réseau de communication.

Plus précisément, les différents moyens identifiés ci-dessus peuvent être incorporés dans un microprocesseur 100, une mémoire morte 101 ("Read-only memory" ou ROM) étant adaptée à mémoriser un programme de
20 proposition d'un service et/ou de test d'accès à un service et/ou de validation d'un message.

Bien entendu, ces dispositifs de proposition d'un service, de test d'accès ou de validation d'un message peuvent être mis en œuvre dans un même ordinateur ou bien dans des stations différentes du réseau de
25 communication.

Une mémoire vive 102 ("Random access memory" ou RAM) est adaptée à mémoriser dans des registres les valeurs modifiées lors de l'exécution du programme de proposition d'un service, ou de test d'accès à un service ou de validation d'un message.

30 Le microprocesseur 100 est intégré à un ordinateur qui peut être connecté à différents périphériques et à d'autres ordinateurs d'un réseau de

communication 10. En particulier, cet ordinateur correspond à un ordinateur serveur S ou un ordinateur client C de ce réseau de communication 10.

Cet ordinateur S, C comporte de manière connue une interface de communication 110 relié au réseau de communication pour recevoir ou
5 transmettre des messages.

L'ordinateur comporte en outre des moyens de stockage de documents, tel qu'un disque dur 106, ou est adapté à coopérer au moyen d'un lecteur de disque 107 (disquettes, disques compacts ou cartes informatiques) avec des moyens de stockage de documents amovibles, tels que des disques
10 7. Ces moyens de stockage fixes ou amovibles peuvent comporter le code des procédés conformes à l'invention.

Ils sont également adaptés à mémoriser un document électronique de description d'un service tel que défini par la présente invention.

A titre de variante, le programme permettant au dispositif de
15 proposition d'un service, de test d'accès ou de validation d'un message peut être stocké dans la mémoire morte 101.

En seconde variante, le programme pourra être reçu pour être stocké comme décrit précédemment par l'intermédiaire du réseau de communication
10 10.

L'ordinateur S, C possède également un écran 103 permettant par
20 exemple de servir d'interface avec un opérateur à l'aide du clavier 104 ou de la souris 105 ou de tout autre moyen.

L'unité centrale 100 (CPU) exécutera alors les instructions relatives à la mise en œuvre de l'invention. Lors de la mise sous tension, les programmes
25 et méthodes relatives à l'invention stockés dans une mémoire non volatile, par exemple la mémoire 101, sont transférés dans la mémoire 102 qui contiendra alors le code exécutable de l'invention ainsi que les variables nécessaires à la mise en œuvre de l'invention.

Le bus de communication 112 permet la communication entre les
30 différents sous-éléments de l'ordinateur ou liés à lui.

La représentation de ce bus 112 n'est pas limitative et notamment le microprocesseur 100 est susceptible de communiquer des instructions à tout sous-élément directement ou par l'intermédiaire d'un autre sous-élément.

Bien entendu, de nombreuses modifications peuvent être apportées
5 aux exemples de réalisation décrits précédemment sans sortir du cadre de l'invention.

REVENDEICATIONS

1. Procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication, caractérisé en ce qu'il comprend
5 une étape d'envoi (E11) d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution dudit service sur le réseau de communication.

2. Procédé de proposition d'un service conforme à la revendication
10 1, caractérisé en ce que ladite fonctionnalité définit des traitements adaptés à produire ou utiliser des données en format XML définies dans une première partie abstraite de document de description d'un service.

3. Procédé de proposition d'un service conforme à la revendication
15 2, caractérisé en ce que la description de ladite fonctionnalité est insérée dans ladite première partie abstraite du document de description d'un service.

4. Procédé de proposition d'un service conforme à l'une des revendications 1 à 3, caractérisé en ce que ledit pré-traitement ou ledit post-traitement est mis en œuvre via un langage de script.

5. Procédé de proposition d'un service conforme à l'une des
20 revendications 1 à 4, caractérisé en ce que ladite fonctionnalité est définie comme une donnée en format XML dans une première partie abstraite de document de description d'un service.

6. Procédé de proposition d'un service conforme à la revendication
25 5, caractérisé en ce que ladite donnée en format XML définissant ladite fonctionnalité est encodée dans une seconde partie concrète du document de description d'un service.

7. Procédé de proposition d'un service conforme à l'une des revendications 1 à 6, caractérisé en ce que la description de ladite fonctionnalité comprend une liste de propriétés supportées par ladite
30 fonctionnalité, lesdites propriétés définissant au moins respectivement :

- le nœud du réseau de communication adapté à exécuter ledit traitement ; et

- le type de traitement.

8. Procédé de proposition d'un service conforme à la revendication 7, caractérisé en ce que ladite fonctionnalité comprend en outre une propriété adaptée à spécifier si ledit traitement est effectué à l'envoi ou à la réception dudit message.

9. Procédé de proposition d'un service conforme à l'une des revendications 7 ou 8, caractérisé en ce que ladite fonctionnalité comprend en outre une propriété adaptée à spécifier le message ou un ensemble de messages sur lequel s'applique ledit traitement.

10. Procédé de proposition d'un service conforme à l'une des revendications 7 à 9, caractérisé en ce que ladite fonctionnalité comprend en outre une propriété adaptée à définir les données produites ou utilisées par ledit traitement, et éventuellement le type desdites données.

11. Procédé de proposition d'un service conforme à l'une des revendications 7 à 10, caractérisé en ce que la description de ladite fonctionnalité comprend une propriété adaptée à spécifier si le traitement à effectuer est obligatoire ou optionnel.

12. Procédé de proposition d'un service conforme à l'une des revendications 7 à 11, caractérisé en ce que pour au moins une propriété supportée par ladite fonctionnalité, la description de ladite fonctionnalité comprend une liste de valeurs attribuables à ladite propriété.

13. Procédé de test d'accès à un service par un ordinateur client d'un réseau de communication, à partir d'un document de description d'un service, caractérisé en ce qu'il comprend les étapes suivantes :

- extraction (E23) d'une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML d'un message échangé lors de l'exécution dudit service sur le réseau de communication ;
- lecture (E24) de la valeur associée à une propriété adaptée à spécifier le nœud du réseau de communication adapté à exécuter ledit traitement ;

- lecture (E24) de la valeur d'une propriété adaptée à spécifier si ledit traitement est obligatoire ou optionnel ; et

- vérification (E25) si ledit traitement est supporté par l'ordinateur client du réseau de communication lorsque ledit traitement est obligatoire et doit être exécuté par ledit ordinateur client du réseau de communication.

14. Procédé de validation d'un message reçu par un ordinateur intermédiaire du réseau de communication, à partir d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format XML du message échangé lors de l'exécution d'un service sur le réseau de communication, caractérisé en ce qu'il comprend les étapes suivantes :

- extraction (E34) d'un traitement dans ledit message reçu ;
- acquisition (E35) d'au moins une valeur impérative associée à une propriété dudit traitement ; et
- vérification (E35) si ladite valeur impérative est incluse dans une liste de valeurs attribuables à une propriété supportée par ladite fonctionnalité décrite dans le document de description d'un service.

15. Procédé conforme à la revendication 14, caractérisé en ce qu'il comprend en outre les étapes suivantes :

- lecture (E36) de la valeur associée à une propriété adaptée à spécifier si le traitement est exécuté avant ou après l'envoi dudit message ; et
- exécution (E37) dudit traitement lorsque ladite valeur est adaptée à spécifier que le traitement doit être exécuté avant l'envoi du message.

16. Dispositif de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication, caractérisé en ce qu'il comprend des moyens d'envoi (100, 101, 102) d'un document de description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou d'un post-traitement de données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication.

17. Dispositif de proposition d'un service conforme à la revendication 16, caractérisé en ce qu'il est incorporé dans :

- un microprocesseur (100) ;

- une mémoire morte (101) adaptée à mémoriser le programme de proposition d'un service ; et

- une mémoire vive (102) comportant des registres adaptés à mémoriser des variables lors de la mise en œuvre du programme.

5 18. Dispositif de test d'accès à un service par un ordinateur client d'un réseau de communication, à partir d'un document de description d'un service, caractérisé en ce qu'il comprend :

- des moyens d'extraction (100, 101, 102) d'une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement du post-traitement de
10 données en format XML d'un message échangé lors de l'exécution du service sur le réseau de communication ;

- des moyens de lecture (100, 101, 102) de la valeur associée à une propriété adaptée à spécifier le terminal du réseau de communication adapté à exécuter le traitement ;

15 - des moyens de lecture (100, 101, 102) de la valeur d'une propriété adaptée à spécifier si le traitement est obligatoire ou optionnel ; et

- des moyens de vérification (100, 101, 102) adaptés à vérifier si le traitement est supporté par l'ordinateur client du réseau de communication lorsque ce traitement est obligatoire est doit être exécuté par l'ordinateur client
20 du réseau de communication.

 19. Dispositif de test d'accès conforme à la revendication 18, caractérisé en ce qu'il comprend des moyens incorporés dans :

- un microprocesseur (100) ;

25 - une mémoire morte (101) adaptée à mémoriser le programme de test d'accès ; et

- une mémoire vive (102) comprenant des registres adaptés à mémoriser les variables modifiées lors de l'exécution dudit programme.

 20. Dispositif de validation d'un message reçu par un ordinateur intermédiaire du réseau de communication, à partir d'un document de
30 description d'un service comprenant une description d'une fonctionnalité mise en œuvre lors d'un pré-traitement ou du post-traitement de données en format

XML du message échangé lors de l'exécution d'un service sur le réseau de communication, caractérisé en ce qu'il comprend :

- des moyens d'extraction (100, 101, 102) d'un traitement dans un message reçu ;

5 - des moyens d'acquisition (100, 101, 102) d'au moins une valeur impérative associée une propriété du traitement ; et

 - des moyens de vérification (100, 101, 102) adaptés à vérifier si la valeur impérative est incluse dans une liste de valeurs attribuables à une propriété supportée par la fonctionnalité décrite dans le document de
10 description d'un service.

21. Dispositif de validation d'un message reçu conforme à la revendication 20, caractérisé en ce qu'il comprend des moyens incorporés dans :

- un microprocesseur (100) ;

15 - une mémoire morte (101) adaptée à mémoriser un programme de validation d'un message reçu ; et

- une mémoire vive (102) comprenant des registres adaptés à mémoriser des variables modifiées lors de l'exécution dudit programme.

22. Ordinateur serveur d'un réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre en œuvre le procédé de proposition d'un service conforme à l'une des revendications 1 à 12.

23. Ordinateur client d'un réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre en œuvre le procédé de test d'accès conforme à la revendication 13.

25 24. Ordinateur d'un réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre le procédé de validation d'un message conforme à l'une des revendications 14 ou 15.

25. Réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre en œuvre le procédé de proposition d'un service
30 conforme à l'une des revendications 1 à 12.

26. Réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre en œuvre le procédé de test d'accès conforme à la revendication 13.

5 27. Réseau de communication, caractérisé en ce qu'il comprend des moyens adaptés à mettre le procédé de validation d'un message conforme à l'une des revendications 14 ou 15.

10 28. Moyen de stockage d'informations, éventuellement totalement ou partiellement amovible, lisible par un système informatique, caractérisé en ce qu'il comprend des instructions pour un programme informatique adaptées à mettre en œuvre le procédé de proposition d'un service conforme à l'une des revendications 1 à 12 lorsque ce programme est chargé et exécuté par le système informatique.

15 29. Moyen de stockage d'informations, éventuellement totalement ou partiellement amovible, lisible par un système informatique, caractérisé en ce qu'il comprend des instructions pour un programme informatique adaptées à mettre en œuvre le procédé de test d'accès conforme à la revendication 13, lorsque ce programme est chargé et exécuté par le système informatique.

20 30. Moyen de stockage d'informations, éventuellement totalement ou partiellement amovible, lisible par un système informatique, caractérisé en ce qu'il comprend des instructions pour un programme informatique adaptées à mettre en œuvre le procédé de validation d'un message conforme à l'une des revendications 14 ou 15, lorsque ce programme est chargé et exécuté par le système informatique.

25 31. Programme d'ordinateur lisible par un microprocesseur, caractérisé en ce qu'il comprend des portions de code logiciel adaptées à mettre en œuvre le procédé de proposition d'un service conforme à l'une des revendications 1 à 12, lorsque ce programme d'ordinateur est chargé et exécuté par le microprocesseur.

30 32. Programme d'ordinateur lisible par un microprocesseur, caractérisé en ce qu'il comprend des portions de code logiciel adaptées à mettre en œuvre le procédé de test d'accès conforme à la revendication 13, lorsque ce programme d'ordinateur est chargé et exécuté par le microprocesseur.

33. Programme d'ordinateur lisible par un microprocesseur, caractérisé ce qu'il comprend des portions de code logiciel adaptées à mettre en œuvre le procédé de validation d'un message conforme à l'une des revendications 14 ou 15, lorsque ce programme d'ordinateur est chargé et
- 5 exécuté par le microprocesseur.

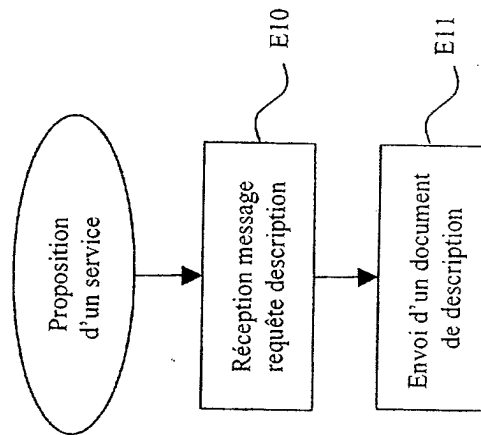


Figure 1

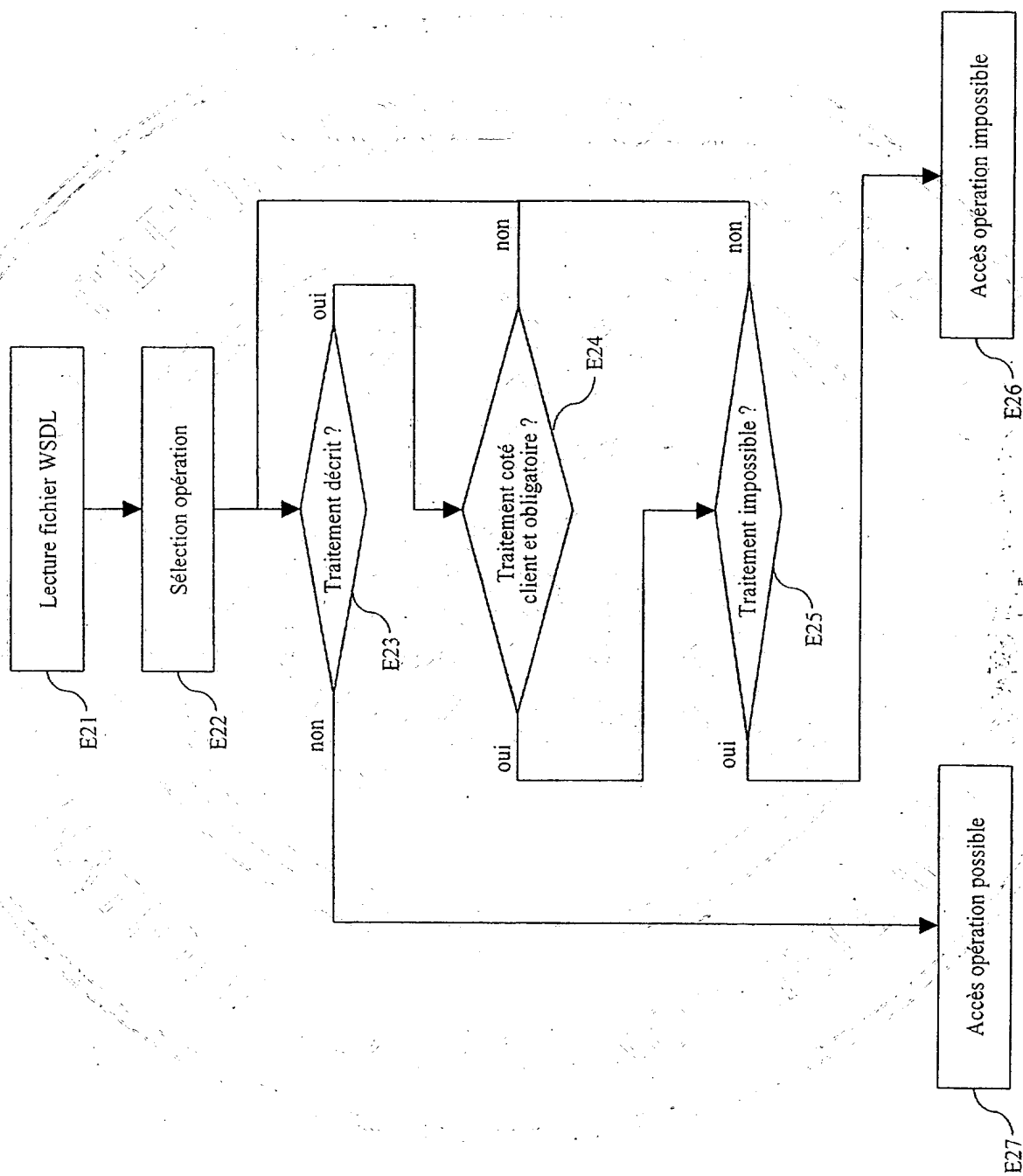


Figure 2

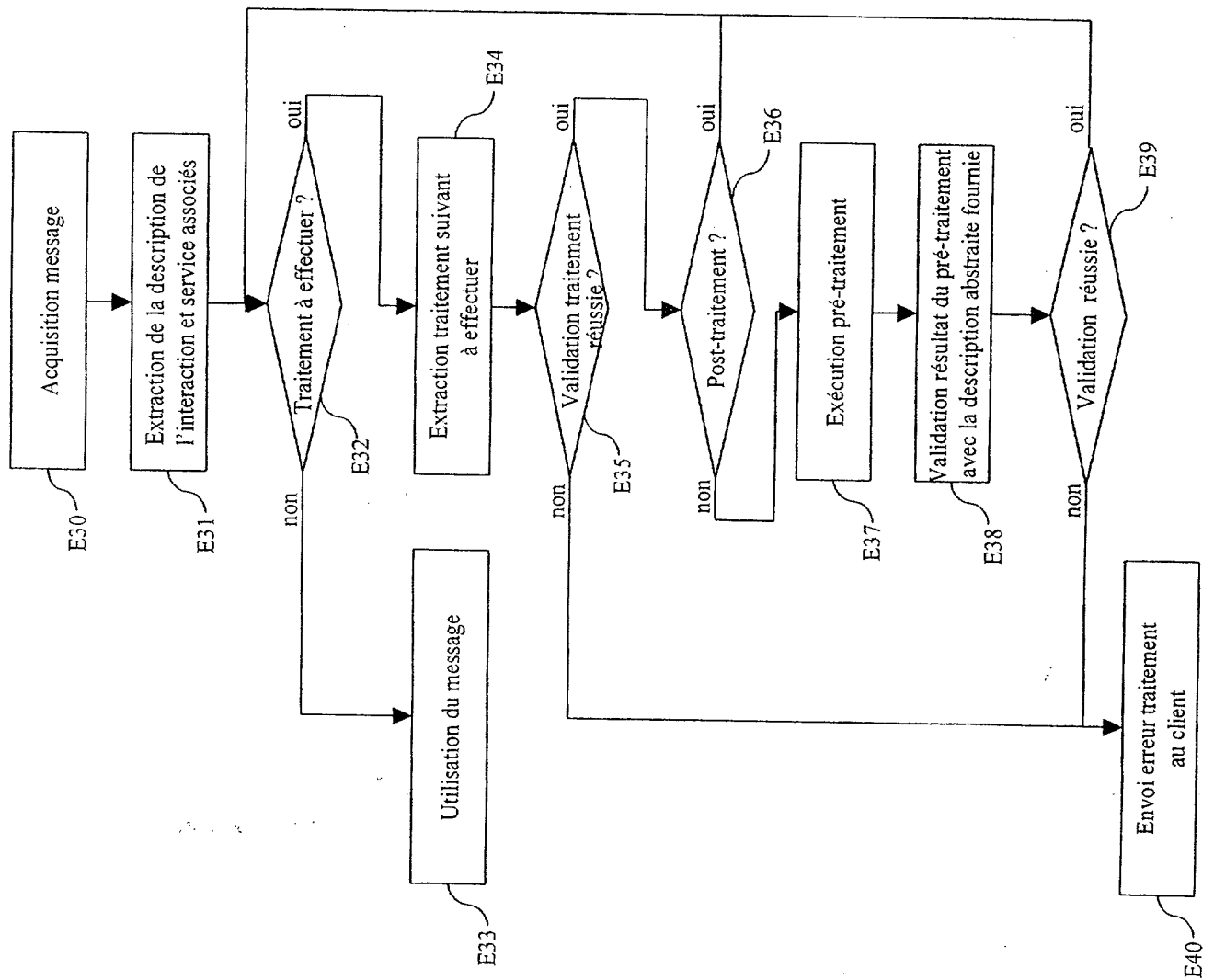


Figure 3

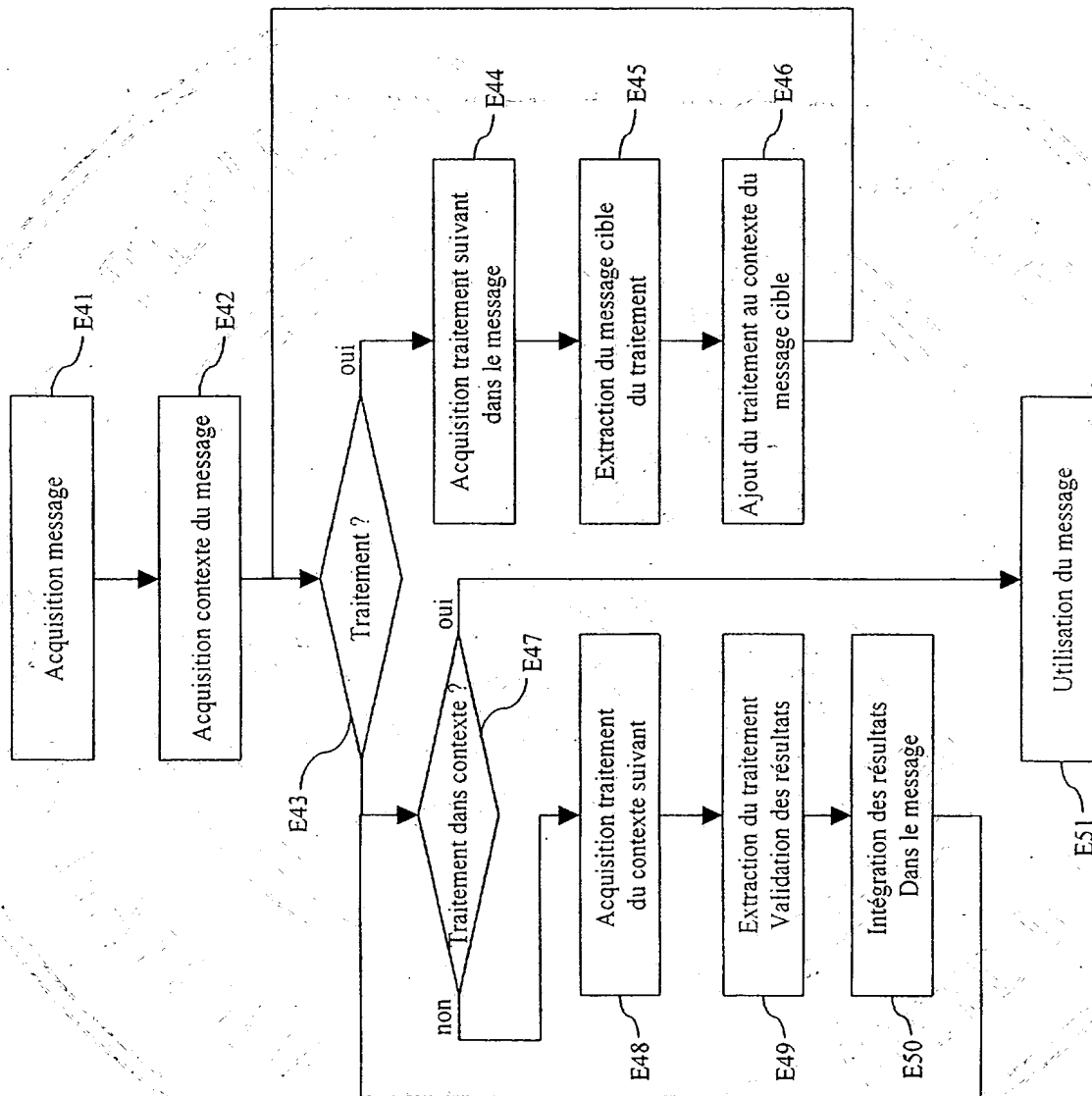


Figure 4

5 / 5

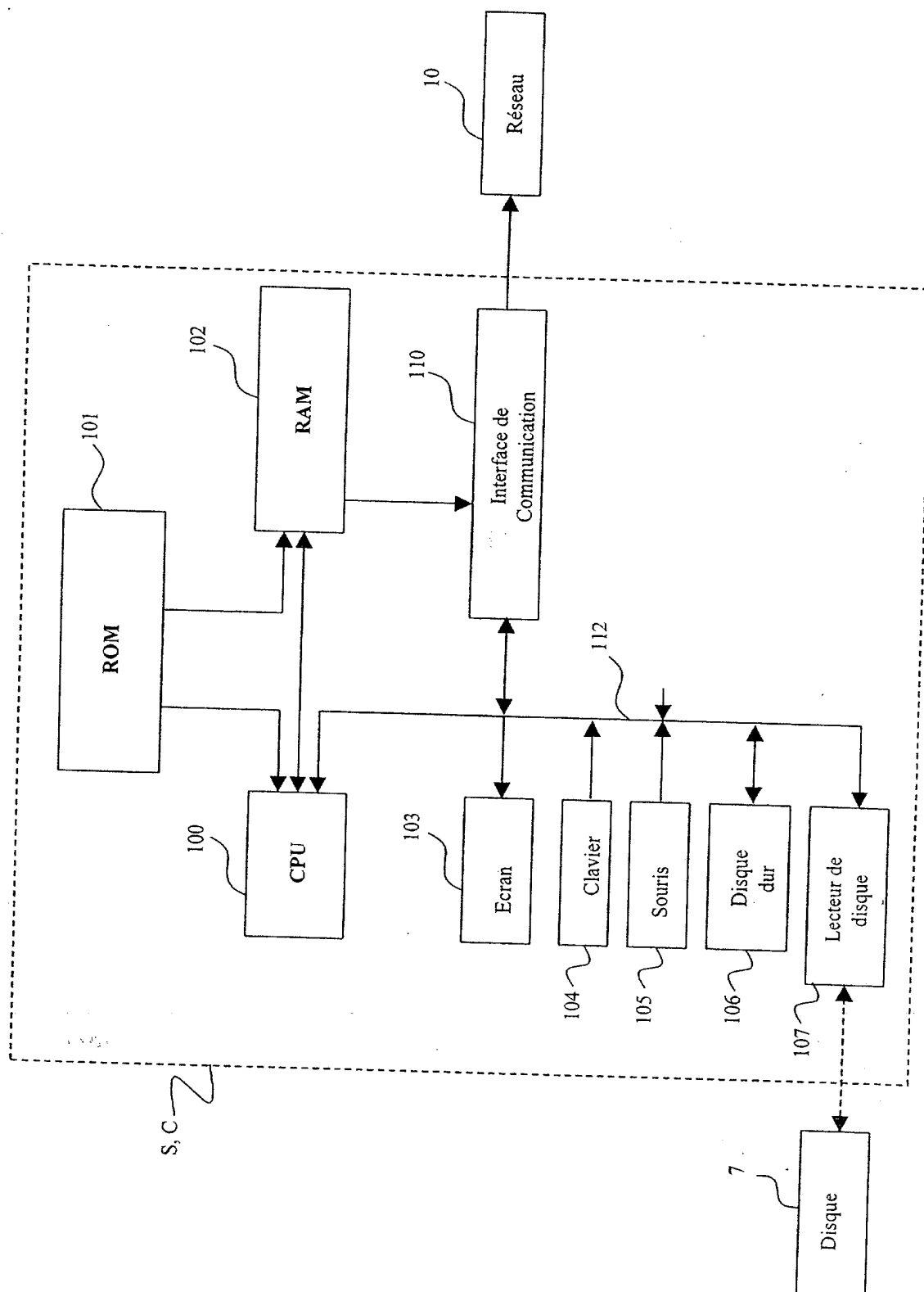


Figure 5

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08

Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° 1. / 1.

(À fournir dans le cas où les demandeurs et
les inventeurs ne sont pas les mêmes personnes)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W / 270601

Vos références pour ce dossier (facultatif)		BIF023427/HST/IM 030246	
N° D'ENREGISTREMENT NATIONAL			
TITRE DE L'INVENTION (200 caractères ou espaces maximum)			
Procédé de proposition d'un service fourni par un ordinateur serveur dans un réseau de communication			
LE(S) DEMANDEUR(S) :			
CANON KABUSHIKI KAISHA			
DESIGNE(NT) EN TANT QU'INVENTEUR(S) :			
1 Nom		FABLET	
Prénoms		Youenn	
Adresse	Rue	Le Pâtis de la Grette	
	Code postal et ville	35390 La DOMINELAIS, France	
Société d'appartenance (facultatif)			
2 Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville	[] [] [] [] []	
Société d'appartenance (facultatif)			
3 Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville	[] [] [] [] []	
Société d'appartenance (facultatif)			
S'il y a plus de trois inventeurs, utilisez plusieurs formulaires. Indiquez en haut à droite le N° de la page suivi du nombre de pages.			
DATE ET SIGNATURE(S) DU (DES) DEMANDEUR(S) OU DU MANDATAIRE (Nom et qualité du signataire)		Le 3 mars 2003 Hélène STANKOFF N°98.0601 SANTARELLI	

THIS PAGE BLANK (USPTO)